AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. - 20. (Canceled)

21. (Currently Amended) A method for managing and transmitting events from a server via a communication link to ~~at least one~~ a client, said method comprising:

logging possible events in a client event service for the purpose of initializing or updating the client,

logging possible events in a server event service for the purpose of initializing or updating the server,

transferring detected events which have been logged from an installation interface to the server event service,

sending requests initiated by the client event service regarding the detected events to the server event service,

transmitting the detected events to the client event service on the basis of a request which has been made to the server event service, and

transmitting events received by the client event service to a client application,

wherein the client application logs a client callback function in the client event service for every event about which the client application is to be notified, and the

client event service uses the communication link to log a corresponding server callback function in the server event service, and

wherein to log the callback functions for an event with which a same event name is associated with the client and with the server, the following steps are performed:

calling, by the client application, a client logging function from the client event service and providing said client logging function with an event name and with a pointer to the client callback function which is to be logged,

logging, by the client logging function, a unique event identifier,

transmitting the event identifier and the event name via the communication link to a server logging function of the server event service,

logging, by the server logging function, a server callback function with the installation interface by transferring the event name,

storing, by the server logging function in a server event table, a data record, which contains at least the event identifier and a pointer to the server callback function which is to be logged,

reporting, by the server logging function, performance of the logging ~~operation~~ to the client logging function of the client event service via the communication link, and

logging, by the client logging function, the client callback function by storing a data record in a client event table, the data record containing at least the event identifier and a pointer to the client callback function which is to be logged.

22.   (Currently Amended) The method as claimed in claim 21, wherein the detected events to be transmitted are detected by a data capture unit in a technical installation and are reported to the installation interface of the server.

23.   (Canceled)

24.   (Previously Presented) The method as claimed in claim 21, wherein after a client callback function has been logged for the first time the client logging function starts a request generator which then makes requests for event transmission to the server event service.

25.   (Previously Presented) The method as claimed in claim 24, wherein the request generator of the client event service makes the requests for event transmission to the server event service cyclically.

26.   (Previously Presented) The method as claimed in claim 24, wherein events are transmitted by performing the following steps:

detecting, by the installation interface, an event which has occurred and calling the server callback function logged for the event,

producing, by the server callback function, an entry describing the event in at least one event queue,

reading, by the server event service, the entry produced in the event queue upon the next request from the client event service for event transmission,

transmitting the entry via the communication link to the client event service,

receiving, by the client event service, the entry,

ascertaining and calling the client callback function logged for the event, and

executing, by the client callback function, a defined action for the

corresponding event in the client application.

27.   (Currently Amended) A system for managing and transmitting events

from a server via a communication link to at least one client, said system comprising:

at least one client, each client configured as a microprocessor coupled to a

memory, each client[[,]] comprising:

a client application[[;]], and

a client event service, for logging possible events for initializing or updating

the client, which uses a communication link to make requests regarding detected

events to a server event service, and transmits received events to the client

application; and

a server configured as a microprocessor coupled to a memory, comprising:

at least one a server event service, which has at least one server logging

function for logging server callback functions, and for logging possible events for the

purpose of initializing or updating the server, and which uses a communication link to

transmit detected events to the client event service on the basis of a request which

has been made to the server event service,

at least one server event table for holding data records which describe a

respective logging operation, which server event table is formed as a hash table and

holds data records which contain at least one event identifier and a pointer to a

server callback function which is to be logged,

at least one event queue for holding entries which describe a respective event, and for transmitting received events to a client application, and

at least one installation interface which transfers detected events ~~which have occurred~~ to the ~~at least one~~ server event service; and

wherein the client application logs a client callback function in the client event service for every event about which the client application is to be notified, and the client event service uses the communication link to log a corresponding server callback function in the server event service,

wherein to log callback functions for an event with which a same event name is associated with the client and with the server, the following is performed:

calling, by the client application, a client logging function from the client event service and providing said client logging function with an event name and with a pointer to the client callback function which is to be logged,

logging, by the client logging function, a unique event identifier,

transmitting the event identifier and the event name via the communication link to a server logging function of the server event service,

logging, by the server logging function, a server callback function with the installation interface by transferring the event name,

storing, by the server logging function in a server event table, a data record, which contains at least the event identifier and a pointer to the server callback function which is to be logged,

reporting, by the server logging function, performance of the logging to the client logging function of the client event service via the communication link, and

logging, by the client logging function, the client callback function by storing a
data record in a client event table, the data record containing at least the event
identifier and a pointer to the client callback function which is to be logged.

28.   (Previously Presented) The method as claimed in claim 21, wherein a
tidying function of the server event service is called which deletes the server event
table and an event queue if the client event service is no longer communicating with
the server event service.

29.   (Previously Presented) The system as claimed in claim 27, wherein
the installation interface is connected to a data capture unit of a technical installation
in order to read in events detected by the data capture unit.

30.   (Previously Presented) The system as claimed in claim 27, wherein
the server event service has at least one server callback function which can be
logged for at least one event and which is called when an event for which it is logged
occurs.

31.   (Previously Presented) The system as claimed in claim 27, wherein
the server event service has, for every client event service with which it
communicates via a communication link, a separate client data record which
respectively contains at least one server event table and at least one event queue.

32.   (Previously Presented) The system as claimed in claim 31, wherein the server event service has a tidying function which deletes the client data record if the associated client event service is no longer communicating with the server event service.

33.   (Canceled)

34.   (Previously Presented) The system as claimed in claim 27, wherein the client event service has at least one client logging function for logging client callback functions, at least one client event table for holding data records which describe the log, and at least one request generator for making cyclic requests for event transmission.

35.   (Previously Presented) The system as claimed in claim 34, wherein the client event table is in the form of a hash table and holds data records which contain at least one event identifier and a pointer to a client callback function which is to be logged.

36.   (Previously Presented) The method as claimed in claim 21, wherein events are transmitted by performing the following steps:

        detecting, by the installation interface, an event which has occurred and calling the server callback function logged for this event,

        producing, by the server callback function, an entry describing the event in at least one event queue,

reading, by the server event service, the entry produced in the event queue

upon the next request from the client event service for event transmission,

transmitting the entry via the communication link to the client event service,

receiving, by the client event service, the entry,

ascertaining and calling the client callback function logged for the event, and

executing, by the client callback function, a defined action for the

corresponding event in the client application.